```
#########################################################################################################
#
FractionToLocation.2 <- function(system, numerator, denominator){
#
#   system = 1  Stern-Brocot
#   system = 2  Bird
#   system = 3  HCS
#   system = 4  Yurramendi-2
#
# Given a fraction numerator / denominator,
# by tracking until the the root (numerator = 1, denominator = 1)
#
# Tracking is related to the Euclid#s algorithm [23]
#
  track <- vector()
#
  while(numerator / denominator != 1){
    if(numerator / denominator < 1){
      if(system == 1  | system == 3)
        denominator <- denominator - numerator
      if(system == 2  | system == 4){
        temp        <- numerator
        numerator   <- denominator - temp
        denominator <- temp
      }
      track <- c(0, track)
    }
    if(numerator / denominator > 1){
        temp        <- numerator
        numerator   <- denominator
        denominator <- temp
      track <- c(1, track)
    }
  }
#
# Tracking back from the root (1,1)
#               to the given fraction
#
  k <- 0
  m <- 0
#
  for(step in 1:length(track)){
    if(track[step] == 1) {
      if(system == 1  | system == 2) k[step+1] <- 2^m[step] - 1 - k[step] ### symmetry
      if(system == 3  | system == 4) k[step+1] <- 2^(m[step]-1) + k[step] ### shifting
      m[step+1] <- m[step]
    }
    else{
      k[step+1] <- k[step]
      m[step+1] <- m[step] + 1
    }
#
  }
  location <- 2^m + k
#
  return(location[length(track)+1])
#
}
#
#########################################################################################################
#
# Examples: Compute location of 9/5 in all four systems
#
FractionToLocation.2(1, 9, 5) ### 55
FractionToLocation.2(2, 9, 5) ### 61
FractionToLocation.2(3, 9, 5) ### 60
FractionToLocation.2(4, 9, 5) ### 51
#
#########################################################################################################
```